

Multi-Expert Routing for Event Graph Completion

Qing Zhang^{1,2} Wei Cai¹ Chao Liang¹ Bang Wang¹ [†]

¹School of Electronic Information and Communications,
Huazhong University of Science and Technology, Wuhan, China

²Shanghai Ace Investment & Development Co., Ltd., Shanghai, China
{zhangqing, caiw, liangchao111, wangbang}@hust.edu.cn

Abstract

The task of *event graph completion* (EGC) aims to predict the missing relations between two events in a heterogeneous *event graph* (EG) to enhance its applicability in downstream tasks. Given that event graphs inherently contain both semantic and structural information, and that the effective aggregation of these two types of information can mutually reinforce their expressive power, we propose a *Multi-Expert Routing* (MER) framework, which leverages the *mixture of experts* (MoE) architecture to dynamically select appropriate experts for collaboratively addressing different key aspects of the EGC task. Specifically, our approach incorporates multiple specialized experts, leveraging prompt learning and a graph attention network, to independently model distinct semantic and structural features within EGs. Furthermore, an *adaptive expert routing* (AER) module is designed to dynamically fuse and select the missing relation predictions from the experts, thereby completing the EGs. Extensive experiments conducted on the EGC-MAVEN dataset demonstrate that our MER framework outperforms the state-of-the-art competitors.

1 Introduction

An *event graph* (EG) is an event-centric knowledge representation paradigm that organizes information around event triggers and their associated arguments (e.g., subject, object, time, and location). Owing to its structured and semantically rich representation of real-world event dynamics, it has been widely applied in various downstream tasks, including question answering, search and recommendation, financial quantitative investment and text generation (Guan et al., 2023; Li et al., 2021; Ugai et al., 2024; Jin et al., 2022).

In this paper, we focus on the task of *event graph completion* (EGC), which aims to infer and com-

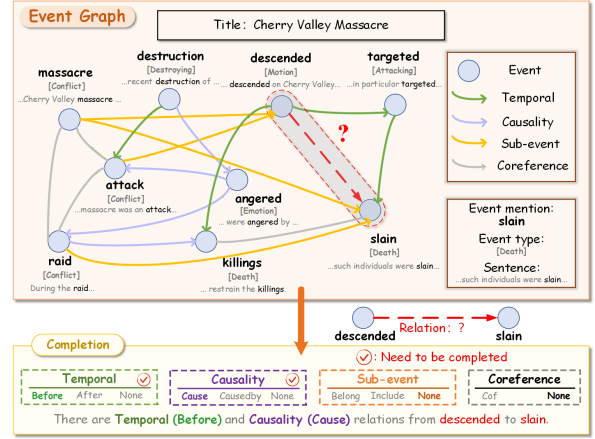


Figure 1: An example of Event Graph Completion (EGC) task.

plete missing relations among events in a heterogeneous event graph (Tang et al., 2021; Wang et al., 2022b; Li et al., 2025b; Hwang et al., 2022). By recovering incomplete structural and semantic dependencies, the EGC improves the overall integrity and quality of the event graph, thereby enhancing its effectiveness in downstream applications.

Figure 1 illustrates an example of EG and the EGC task. In this EG, each event node is associated with the event mention word(s), event type, and the sentence to which it belongs. The relations between events include four types, namely, the 'temporal', 'causal', 'sub-event', and 'coreference'. Among these relation types, the 'coreference' is undirected, whereas the others are directed. It is worth noting that multiple relations may coexist between the same pair of event nodes in a heterogeneous event graph; consequently, this EG can be decomposed into four homogeneous subgraphs, each containing only one type of relation. Given an EG and a pair of events (e.g., the events "descended" and "slain"), the EGC task aims to infer the missing relations between the two events. Specifically, this involves determining whether one or more relations exist

[†]Corresponding author.

between the event pair, and if they do, identifying their specific directions (e.g., a 'before' relation and a 'cause' relation in this example).

The pre-trained language models (PLMs) have demonstrated remarkable performance and have become foundational components in a wide range of downstream natural language processing tasks (Liu et al., 2024; Zhu et al., 2024; Du et al., 2024; Min et al., 2023). For instance, the TEPrompt is a task-enlightened prompt learning framework that integrates knowledge learned from three related tasks of implicit discourse relation recognition (Xiang et al., 2023). Despite their success, most PLM-based prompt learning approaches are designed for sequential text inputs and thus are not naturally suited for graph-structured data. This limitation restricts their direct application to structured knowledge representations such as event graphs. To leverage the rich semantic knowledge embedded in PLMs while accommodating graph structures, we adopt a graph prompt learning paradigm that jointly encodes both the semantic content and structural dependencies of an EG for missing event relation prediction.

Recently, the *mixture of experts* (MoE) architecture in large language models offers significant advantages by utilizing specialized experts to tackle different aspects of complex tasks (Masoudnia and Ebrahimpour, 2014; Zhou et al., 2022; Shazeer et al., 2017; Mu and Lin, 2025). For example, the MOOSE, a robust automated essay scoring framework, employs three specialized experts to assess essay quality by integrating multi-aspect information from both prompts and essays (Chen et al., 2025). However, as the MoE architecture is typically designed and tailored for specific task scenarios, its direct application to the EGC task poses non-trivial challenges. To fully leverage the strengths of the MoE architecture for the EGC task, we adopt a routing mechanism that dynamically selects and weights the experts based on the given query event pair.

In this paper, we propose a **Multi-Expert Routing** (MER) framework for the EGC task. Our approach incorporates multiple specialized experts, leveraging both prompt learning and a graph attention network, to capture the semantic and structural features of event graphs. To effectively integrate the outputs from different experts, we design an *Adaptive Expert Routing* (AER) module. For each query event pair, the AER module dynamically selects the most relevant experts and adaptively as-

signs query-specific weights to yield the final prediction. Experiments on the EGC-MAVEN dataset validate the effectiveness of our MER framework, which achieves new state-of-the-art results. Ablation studies also demonstrate the effectiveness of each module in our framework. The discussion of related work is deferred to Appendix A.

2 Methodology

2.1 Preliminaries

As the task initialization, we have an *event graph* $\mathcal{G}$ constructed from a document. Each event node $e_i \in \mathcal{G}$ is associated with the event mention e_i^m , event type e_i^t and the sentence containing the event mention e_i^s , all in natural language. An *event triplet* is denoted as $\tau_{ij} = (e_i, \mathcal{R}_{ij}, e_j)$, where $\mathcal{R}_{ij}$ denotes the set of relations between an event e_i and another event e_j . In this paper, we use the following words for indicating each relation type: The 'temporal' relation is indicated by {before, after, none}, 'causal' relation by {cause, caused_by, none}, 'sub-event' relation by {belong, include, none}, and 'coreference' relation by {coreference, none}. For a *query event pair* (e_h, e_t) , the task of *event graph completion* is to predict the missing relations $\mathcal{R}_{ht}$ between the head and tail event of the query.

In this paper, we use a [mask] token to compose the *query event triplet* $\tau_{ht} = (e_h, [\text{mask}], e_t)$, where the [mask] stands for the relations $\mathcal{R}_{ht}$ to be predicted. We design the MER framework to encode the [mask] into a tensor $\mathbf{R}_{ht}$ that consists of four prediction probability vectors $\{\mathbf{r}_t, \mathbf{r}_c, \mathbf{r}_s, \mathbf{r}_f\}$, each used for predicting the corresponding relation type. For example, the probability vector $\mathbf{r}_t$ is used to predict the 'temporal' relation.

Figure 2 illustrates the proposed MER framework, which utilizes a MoE architecture to dynamically select and combine multiple experts for relation prediction. It consists of two components: The *multiple expert modeling* (MEM) generates individual relation predictions by leveraging two kinds of modeling techniques, i.e., prompt learning and graph attention network; The AER selects experts for predictions and aggregates the selected experts' predictions to yield the final result.

2.2 Multi-Expert Modeling

To capture the semantic and structural information of an event graph, we design two kinds of experts: One is implemented with prompt learning with em-

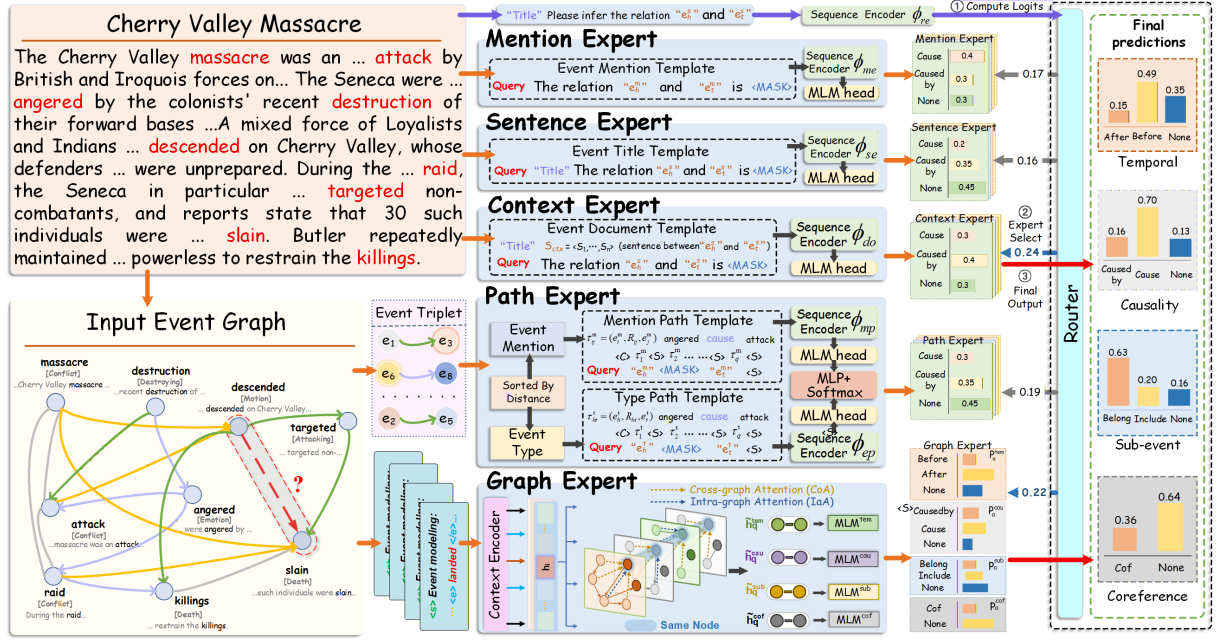


Figure 2: Illustration of our proposed Multi-Expert Routing (MER) framework.

phasis on semantic information, the other is implemented with a graph attention network integrating both semantic and structural information.

2.2.1 Mention Expert

The most straightforward semantic modeling is to encode only the event mentions for prediction. Despite simplicity, sometimes mention words may be enough to infer a relation, for they are appropriately encoded with a language model with enough world knowledge. For example, the event mention words "earthquake" and "deaths" exhibit a clear causal dependency that can be readily inferred from general world knowledge.

The *mention expert* is designed to capture such intrinsic semantic associations by prompting a *mention encoder* ϕ_{me} (e.g., a pre-trained language model, RoBERTa) with the following mention template (sketch) $\mathbb{T}_{me}$:

$$\mathbb{T}_{me} = \text{The relation } e_h^m \text{ and } e_t^m \text{ is [mask] ?}$$

The e_h^m and e_t^m , respectively, are the mentions of the head and tail event of the query event pair. We note that the mention encoder ϕ_{me} is only for the mention expert and its parameters are updated during training. After the sequence encoding, we use a *masked language model* (MLM) head to estimate the probability of each word in its vocabulary for the [mask] tokens and output the relation prediction $\mathbf{R}_{ht}^{me}$.

2.2.2 Sentence Expert

Words could be better understood within a sentence context. We also design a *sentence expert* that includes the sentences containing the mention e_h and e_t for relation prediction. Besides, the title of a document summarizes the main topic of the entire document, providing some background information for comprehension. So we also incorporate the document title to help infer event relations. Specifically, we design the following sentence template (sketch) $\mathbb{T}_{se}$ for prompting the *sentence encoder* ϕ_{se} of the sentence expert:

$$\mathbb{T}_{se} = \text{'Title' The relation } e_h^s \text{ and } e_t^s \text{ is [mask] ?}$$

where 'Title' is the document title, and e_h^s and e_t^s , respectively, the sentence containing the head event and tail event. Similar to the mention expert, the sentence expert uses an MLM to output the relation prediction $\mathbf{R}_{ht}^{se}$.

2.2.3 Context Expert

We further investigate whether more contextual information could help an expert reason about the missing relations. However, as the full document is often too long and beyond the input window limit, we consider using the sentences between the event e_h and e_t as contextual information. Let $S_{ctx} = \langle s_1, \dots, s_n \rangle$ denote the sequence of sentences between e_h and e_t . We design the following context template (sketch) for prompting the *context*

encoder ϕ_{ce} of the context expert:

$$\mathbb{T}_{ce} = \text{'Title'} S_{ctx} \text{The relation } e_h^s \text{ and } e_t^s \text{ is [mask] ?}$$

The context expert applies an MLM to output the relation prediction $\mathbf{R}_{ht}^{ce}$.

2.2.4 Path Expert

The aforementioned experts focus on event semantic information without considering the structural information of the event graph. We argue that the known relations between other event pairs can help to infer the relations of the query event pair. For event triplets with underlying graph-structure, we propose to first order them according to their distances to the query event pair. Specifically, for an *event mention triplet* $\tau_{ij}^m = (e_i^m, \mathcal{R}_{ij}, e_j^m)$, its distance to the query triplet τ_{ht} is defined as $\min\{d_{ih}, d_{it}, d_{jh}, d_{jt}\}$, where $d_{..}$ denotes the shortest-path between two event nodes in the event graph, regardless of the edge type and direction.

Besides ordering event mention triplets, we also order *event type triplets* into a sequence as follows. For an event type triplet $\tau_{ij}^t = (e_i^t, \mathcal{R}_{ij}, e_j^t)$, its distance to the $(e_h^t, \mathcal{R}_{ht}, e_t^t)$ is computed the same as the mention triple distance, with the exception that if two event types are the same, their distance is set to zero. We note that the two simple ordering mechanisms are mainly for providing a sequential input to the sequence encoder. As each triplet contains the relation type, we expect that the sequence encoder of the *path expert* can itself distinguish different relation types in prediction.

We use some PLM-specific tokens [CLS] and [SEP] to form a mention path template $\mathbb{T}_{pe}^m$ and a type path template $\mathbb{T}_{pe}^t$ as follows:

$$\mathbb{T}_{pe}^m = [\text{CLS}] \tau_1^m [\text{SEP}] \tau_2^m \dots [\text{SEP}] \tau_m^m [\text{SEP}] \tau_q^m [\text{SEP}]$$

$$\mathbb{T}_{pe}^t = [\text{CLS}] \tau_1^t [\text{SEP}] \tau_2^t \dots [\text{SEP}] \tau_m^t [\text{SEP}] \tau_q^t [\text{SEP}]$$

where $\tau_1^m \dots \tau_m^m$ ($\tau_1^t \dots \tau_m^t$) is the ordered sequence of all event mention triplets (event type triplets) in the event graph, and $\tau_q^m = (e_h^m, [\text{mask}]^m, e_t^m)$ is the query event mention triplet and $\tau_q^t = (e_h^t, [\text{mask}]^t, e_t^t)$ is the query event type triplet. Notice that $[\text{mask}]^m$ and $[\text{mask}]^t$ are encoded by two different sequence encoders, and their encodings are denoted by $\mathbf{R}^{mp}$ and $\mathbf{R}^{ep}$.

We next fuse the two encodings as the output of the path expert. Specifically, we concatenate $\mathbf{R}^{mp}$ and $\mathbf{R}^{ep}$, and pass them through a multi-layer perceptron followed by a softmax function as the

prediction output $\mathbf{R}_{ht}^{pe}$ of the path expert, i.e.,

$$\mathbf{R}_{ht}^{pe} = \text{Softmax}(\mathbf{W}^{pe}(\mathbf{R}^{mp} || \mathbf{R}^{ep})),$$

where $\mathbf{W}^{pe}$ is a trainable matrix.

2.2.5 Graph Expert

The aforementioned experts primarily rely on training PLM-based sequence encoders for relation prediction by linearizing local or multi-hop textual information into prompt templates. We next design a *graph expert* to explicitly model the global structural interactions among events on the event graph. To this end, we adopt the *graph attention networks* (GAT) for the graph expert. Specifically, we decompose the heterogeneous event graph $\mathcal{G}$ into four homogeneous graphs $\{\mathcal{G}_t, \mathcal{G}_c, \mathcal{G}_s, \mathcal{G}_f\}$, each containing all event nodes but only with edges of the same relation type. For example, the homogeneous 'temporal' event graph $\mathcal{G}_t$ contains only the edges of the temporal relations.

Each node in $\mathcal{G}$ is first initialized with a PLM-based semantic encoder with the input of the event sentence. We next design a trainable matrix for each homogeneous graph to transform the heterogeneous graph initialization $\mathbf{h}_i$ into the homogeneous graph-specific initialization. Take the $\mathcal{G}_t$ for example, $\mathbf{h}_i^t = \mathbf{W}^t \mathbf{h}_i$. The standard message passing and aggregation process of the graph attentive network is then performed on each homogeneous graph to obtain the final representation for each node, e.g., the $\tilde{\mathbf{h}}_i^t$ of node i in $\mathcal{G}_t$.

In each homogeneous graph, we use the representations of the query head node and tail node for relation prediction. Take the 'temporal' graph $\mathcal{G}_t$ for example. We compute $\mathbf{r}_q^t = \tilde{\mathbf{h}}_h^t - \tilde{\mathbf{h}}_t^t$ and input $\mathbf{r}_q^t$ into a masked language head followed by a softmax layer for only predicting the 'temporal' relation. We use $\mathbf{R}_{ht}^{ge}$ to denote the predicted relations by the graph expert.

2.3 Adaptive Expert Routing

Since different experts specialize in different aspects of event graphs, a static combination strategy may not fully leverage their complementary strengths. We introduce an expert router to adaptively select and weight experts based on the query event pair. Specifically, we design the routing template $\mathbb{T}_r$ for the query event pair:

$$\mathbb{T}_r = \text{'Title'} \text{Infer the relation } e_h^s \text{ and } e_t^s.$$

and feed this template into a *router encoder* ϕ_{re} to obtain the hidden state $\mathbf{h}$ of the start token as the template representation.

The template representation $\mathbf{h}$ is passed through a linear transformation layer, followed by a softmax operation to obtain the routing probabilities $\mathbf{P}$ for each expert.

$$\mathbf{P} = \text{softmax}((\mathbf{w} \cdot \mathbf{h} + \mathbf{b} + \epsilon \cdot \mathcal{N}(0, 1))/t),$$

where $\mathbf{w}$ and $\mathbf{b}$ denote the learnable weight matrix and the bias vector used in the linear transformation layer, respectively, t is a temperature parameter used to control the sharpness of the distribution, and ϵ is a hyperparameter to prevent routing collapse.

We select the top- k experts according to their probabilities and integrate the outputs of the selected experts $\mathcal{K}$ as the final prediction $\mathbf{R}$:

$$\mathbf{R} = \sum_{k \in \mathcal{K}} \mathbf{P}^k \cdot \mathbf{R}_{\text{ht}}^k,$$

where $\mathbf{R}_{\text{ht}}^k$ denotes a selected expert from $\{\mathbf{R}_{\text{ht}}^{\text{me}}, \mathbf{R}_{\text{ht}}^{\text{se}}, \mathbf{R}_{\text{ht}}^{\text{ce}}, \mathbf{R}_{\text{ht}}^{\text{pe}}, \mathbf{R}_{\text{ht}}^{\text{ge}}\}$.

2.4 Training Strategy

We employ a two-stage training strategy to train experts and the router independently. The first is the independent expert training, where each expert is optimized individually; The second is adaptive router training, where the router is trained while keeping expert parameters frozen.

For each expert, we use a constrained masked language modeling objective. Specifically, we restrict the output to a predefined set of answer tokens corresponding to each event relation (e.g., mapping the 'temporal' relation to tokens $\{\text{'before'}, \text{'after'}, \text{'none'}\}$), reformulating the relation prediction as cloze-style prediction. Formally, each expert is trained independently by minimizing the following cross-entropy loss $\mathcal{L}_{\text{expert}}$:

$$\mathcal{L}_{\text{expert}} = -\log P(\mathbf{y} \mid \mathbb{T}),$$

where $P(\mathbf{y} \mid \mathbb{T})$ denotes the probability computed by the expert regarding the ground truth $\mathbf{y}$.

After training each expert, we freeze all expert parameters and train the adaptive expert router. Specifically, we detach the outputs of each expert before integration, and we further apply an inverse-frequency weight based on the ratio of the number of total samples N to the number of ground-truth samples N_f to mitigate the imbalance of event relation types. Formally, the router is trained by

Prompt of LLMs

Description: We annotated the events and their relationships in the given document, and constructed an event graph with the events as nodes and the relations between them as edges. There are four types of relations: Sub-event, Temporal, Causal, Coreference.

Sentence:
Expedition: The Expedition of the Thousand was an event of the Italian Risorgimento that took place in 1860.
Sailed: A corps of volunteers sailed from Quarto.

Brought: The expedition was a success and concluded with a plebiscite that brought Naples and Sicily into the Kingdom of Sardinia.

Structure:
 Sailed Before landed [SEP] landed Before conquer [SEP] Expedition
 Cof expedition [SEP] Expedition Include Sailed [SEP]

Question:
 Please determine if there is a **missing** relation between Event1 Expedition and Event2 Brought.

Note:

- Sub-event: This includes the relations 'Belong', 'Include', or 'None'.
- Temporal: This includes the relations 'Before', 'After', or 'None'.
- Causal: This includes the relations 'Cause', 'CausedBy', or 'None'.
- Coreference: This includes the relation 'Cof' (Coreference) or 'None'.

Respond with a single word or phrase indicating the missing relation, or 'None' if no relation is missing.

Figure 3: Prompt input for the LLMs.

minimizing the cumulative weighted cross-entropy loss $\mathcal{L}_{\text{router}}$ across four event relations $\mathcal{F}$:

$$\mathcal{L}_{\text{router}} = \sum_{f \in \mathcal{F}} (N/N_f) \cdot \mathcal{L}_{\text{CE}}(\mathbf{R}_f, \mathbf{y}_f),$$

where $\mathcal{L}_{\text{CE}}$ denotes the cross-entropy loss.

3 Experiments

3.1 Experiment Setup

Dataset: We conduct experiments on EGC-MAVEN dataset (Liang et al., 2026). Each instance is a document containing event mentions annotated with semantic types. The dataset also provides coreference annotations that link mentions referring to the same event, as well as inter-event relation annotations, including temporal, causal, and sub-event relations.

Implementation Details: We use RoBERTa-base (Liu et al., 2019) as the text encoder via the HuggingFace Transformers library. All experiments are implemented in PyTorch and run on a NVIDIA RTX 3090 GPU with CUDA.

Metrics: We evaluate the model performance using Precision (P), Recall (R), and macro-F1 (F1). More details about the dataset, implementation and metrics can be found in Appendix B.

We employ some advanced knowledge graph completion models as baselines for the EGC task, including HINGE (Rosso et al., 2020), CONGLR (Lin et al., 2022), APST (Su et al., 2024),

Model	Sub-event			Causality			Temporal			Coreference			Average		
	R	P	F1	R	P	F1	R	P	F1	R	P	F1	R	P	F1
ChatGLM3-6B (Team GLM, 2024)	18.0	8.7	10.3	47.6	13.4	20.9	24.6	7.1	11.1	46.2	25.3	32.7	34.1	13.6	18.7
Llama3-8B (Grattafiori et al., 2024)	47.6	11.7	14.4	46.6	13.9	19.8	41.1	16.1	21.4	91.3	15.8	27.0	56.7	14.4	20.6
GPT-3.5-turbo (Ouyang et al., 2022)	38.7	12.0	18.2	47.6	14.6	22.3	41.0	18.4	24.7	89.4	20.5	33.4	54.2	16.4	24.6
DeepSeek-V3.2 (DeepSeek-AI et al., 2025)	48.2	19.0	22.2	34.9	12.4	17.3	42.8	31.2	33.8	40.0	34.7	30.3	41.5	24.3	25.9
GPT-4o-mini (OpenAI, 2024)	46.0	25.2	22.7	35.9	16.9	22.9	41.2	<u>40.9</u>	<u>39.0</u>	42.7	38.5	38.0	41.5	30.4	30.7
HINGE (Rosso et al., 2020)	80.7	15.4	25.6	69.1	16.6	26.4	55.3	16.0	24.0	89.6	33.3	48.5	<u>73.7</u>	20.3	31.1
ConGLR (Lin et al., 2022)	63.5	36.3	45.7	52.7	34.3	41.6	47.4	25.0	32.7	82.7	74.5	78.4	61.6	42.5	49.6
SIEG (Shi et al., 2024)	33.7	19.2	24.2	45.8	14.7	22.1	36.9	17.7	23.8	88.0	53.0	66.1	51.1	26.1	34.1
APST (Su et al., 2024)	67.9	22.1	32.8	<u>72.8</u>	25.8	38.0	<u>55.9</u>	23.7	32.9	95.1	59.0	72.8	72.9	32.6	44.1
PLAF (Liang et al., 2026)	60.2	<u>55.7</u>	<u>57.9</u>	52.2	<u>50.4</u>	<u>51.2</u>	44.4	29.9	34.3	97.9	<u>77.8</u>	<u>86.7</u>	63.7	<u>53.5</u>	<u>57.5</u>
MER	<u>73.0</u>	59.1	65.2	76.6	52.1	62.0	59.0	42.8	50.2	<u>97.0</u>	79.4	87.3	76.4	58.4	66.2

Table 1: Overall results of comparison models on the EGC-MAVEN dataset. The best results are in bold, and the sub-optimal results are underlined.

SIEG (Shi et al., 2024), and PLAF (Liang et al., 2026). Besides, we also evaluate the effectiveness of large language models (LLMs) on the EGC task, including ChatGLM3-6B (Team GLM, 2024), Llama3-8B (Grattafiori et al., 2024), GPT-3.5-turbo (Ouyang et al., 2022), GPT-4o-mini (OpenAI, 2024), and DeepSeek-V3.2 (DeepSeek-AI et al., 2025). All LLM baselines use the same prompt template illustrated in Figure 3. The details of these competitors are provided in Appendix C.

3.2 Main Results

Table 1 compares the event graph completion performance between our MER framework and the competitors on the EGC-MAVEN dataset. It is observed that our MER achieves competitive performance across multiple metrics on the EGC-MAVEN dataset, demonstrating notable improvements over baseline models in four relation completions. In particular, MER improves the F1-scores of Sub-event, Causality, Temporal, and Coreference relation completions by 7.3, 10.8, 11.2, and 0.6 percentage points over the second-best results, respectively. The results validate our design motivations and objectives. The superior performance of MER stems from its MoE architecture. Specifically, our prompt-based experts provide rich semantic representations by leveraging the deep contextual knowledge of PLMs; while the Graph Expert boosts global structural reasoning by capturing complex intra- and inter-relation dependencies. Finally, the Adaptive Expert Routing dynamically selects and fuses the outputs of each expert, allowing the framework to adapt to different relation types and graph structures.

We note that the LLMs in zero-shot learning struggle to outperform the other well-trained approaches. Notably, this performance gap persists even when scaling up the LLMs parameter count, which indicates that the EGC is still a challenging task for the LLMs due to the complexity of graph structures and event semantics. Consequently, the superiority of our MER framework is not solely derived from the fine-grained modeling of structural and semantic information, but relies on its MoE architecture, which dynamically selects the most relevant experts for each event pair.

3.3 Ablation and Parameter Results

Module Ablation We conduct module ablation experiments to evaluate the contributions of each component in MER. The ablation strategies are defined as follows: (1) Single Expert: uses only one specific expert; (2) Individual Expert Ablation: removes each of the five experts from the full framework one at a time; (3) w/o Router: replaces the AER module with a simple majority voting.

Table 2 provides the ablation study results, where an expert or router is removed from the full MER framework. We can observe that the full MER framework outperforms all single experts, demonstrating the necessity of the MoE architecture. Besides, even when a top-performing expert is removed, the framework still maintains performance comparable to the best single expert.

Furthermore, each expert and the router contribute to the F1 performance improvements of the full framework. Consistent with their strong independent performance, the largest performance drops happen when removing the Context and Path

Model	Sub-event			Causality			Temporal			Coreference			Average		
	R	P	F1	R	P	F1	R	P	F1	R	P	F1	R	P	F1
Only Mention Expert	41.5	14.9	21.3	53.2	20.6	29.5	45.0	19.0	26.6	84.0	33.6	48.0	55.9	22.0	31.4
Only Sentence Expert	66.3	35.7	46.4	73.2	37.3	49.4	48.5	34.0	39.8	<u>97.7</u>	69.7	81.3	71.4	44.2	54.2
Only Context Expert	65.1	40.3	49.5	<u>79.4</u>	42.6	55.4	<u>62.1</u>	41.0	49.3	93.7	79.3	85.9	75.1	50.8	60.0
Only Path Expert	73.0	56.3	63.5	64.6	46.4	54.0	48.0	34.3	40.0	98.8	69.1	81.4	71.1	51.5	59.7
Only Graph Expert	80.4	20.0	32.0	71.2	26.8	38.9	52.5	22.7	31.7	78.6	64.9	71.1	70.7	33.6	43.4
w/o Mention Expert	74.1	56.6	<u>64.1</u>	79.3	<u>49.8</u>	<u>61.2</u>	59.8	<u>41.5</u>	48.9	96.7	<u>79.7</u>	<u>87.4</u>	<u>77.5</u>	<u>56.9</u>	<u>65.4</u>
w/o Sentence Expert	73.9	<u>56.7</u>	<u>64.1</u>	78.6	49.4	60.7	59.1	40.9	48.3	97.0	79.3	87.2	77.2	56.6	62.6
w/o Context Expert	73.0	56.6	63.7	64.7	46.4	54.1	48.0	34.3	40.0	98.8	69.1	81.4	71.1	51.6	59.8
w/o Path Expert	64.8	40.5	49.8	78.9	43.1	55.7	61.8	40.8	49.2	94.2	78.8	85.8	74.9	50.8	60.1
w/o Graph Expert	<u>75.2</u>	48.0	58.6	80.1	47.9	60.0	63.2	<u>41.5</u>	<u>50.0</u>	95.6	82.6	88.6	78.5	55.0	64.3
w/o Router	73.3	46.3	56.4	73.3	42.0	52.4	55.0	33.0	41.2	95.5	78.9	86.4	74.3	50.0	59.1
MER	73.0	59.1	65.2	76.6	52.1	62.0	59.0	42.8	50.2	97.0	79.4	87.3	76.4	58.4	66.2

Table 2: Experiment results of ablation study on the EGC-MAVEN dataset.

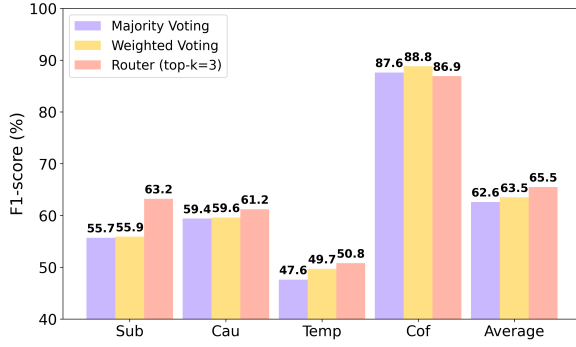


Figure 4: Results of different routing strategies. (top-k=3)

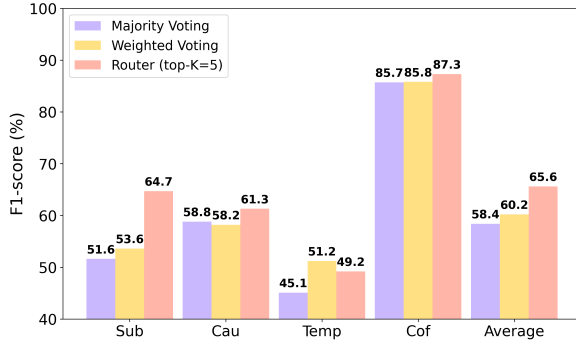


Figure 5: Results of different routing strategies. (top-k=5)

experts, which highlights the importance of contextual descriptions and relational paths for EGs. It is interesting to note that removing the router also leads to a notable decline, indicating that our MER framework is not simply from the strength of individual experts, but rather from the ability of the router to adaptively integrate their predictions.

Effect of Routing Strategies We compare the AER module with majority voting and weighted

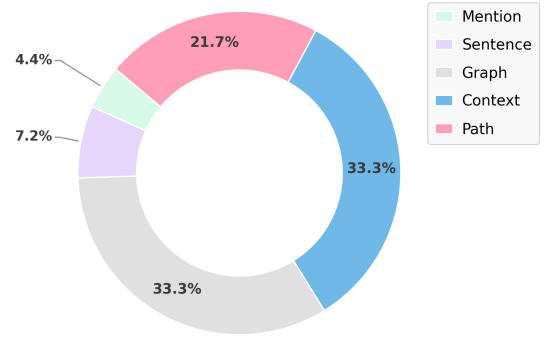


Figure 6: Distribution of different experts on test set.

voting under two settings: (1) the AER module is configured with top-k=3, while the voting strategies employ the three experts most frequently selected by the AER module; (2) the AER module is configured with top-k=5, while the voting strategies aggregate predictions from all five experts.

Figure 4 presents the experimental results under the first setting. We can observe that the AER module achieves the highest average F1-scores. We attribute this superior performance to its dynamic routing mechanism. While static voting strategies rely on a fixed set of experts and treat all inputs equally, our AER module recognizes that different experts specialize in different event pairs, and adaptively selects the most suitable experts for each input event pair.

We further evaluate the routing strategies under the second setting, and the results are illustrated in Figure 5. The AER module still outperforms the voting strategies. Static voting strategies assign fixed weights to each expert regardless of the

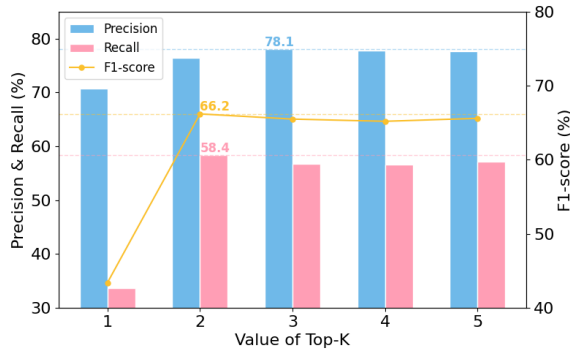


Figure 7: Impact of Top-k.

Model	Sub.	Cau.	Tem.	Corf.	Ave.
MER BERT	61.8	51.2	36.2	85.1	58.6
MER ERNIE	63.8	56.9	42.1	88.2	62.8
MER DeBERTa	51.6	57.2	45.6	86.7	60.3
MER RoBERTa	65.2	62.0	50.2	87.3	66.2

Table 3: Performance of different PLMs (F1-score %).

input. In contrast, the AER module dynamically adjusts the weights of different experts for each query event pair, allowing the framework to emphasize more relevant experts while down-weighting less suitable ones. These comparisons again validate the effectiveness of our AER module.

Expert Selection Distribution The MoE architecture is designed to dynamically route varying inputs to distinct specialized experts. Figure 6 plots the selection distribution of different experts on the test dataset. It is observed that Context and Path experts dominate the selection, while the remaining experts are sparsely activated. This suggests that contextual information and relational paths are more important for event reasoning. Furthermore, this varied distribution confirms that the router dynamically selects experts, without suffering from routing collapse. The expert selection distribution on the training and validation sets can be found in Appendix D.

Parameter Study of Top-k The top-k parameter helps to control the number of experts by the router. Figure 7 presents the prediction performance against the values of top-k. Results indicate that top-k=2 is the best choice for our MER. When top-k=1, the model selects too few experts, focusing only on the local information of the event graph and underutilizing the potential of the MoE architecture. Conversely, when top-k>2, an excessive number of experts may cause interference among them and introduce more unnecessary noise.

Study of different PLMs To explore the impact

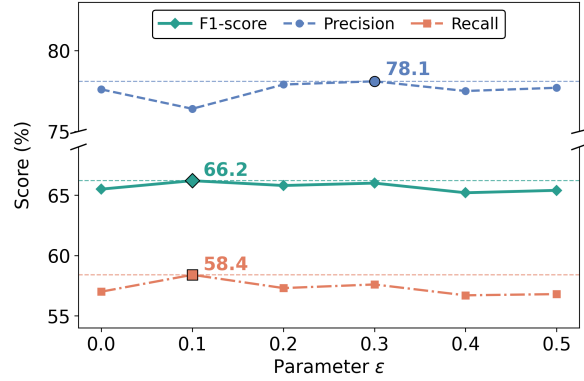


Figure 8: Impact of noise ϵ .

of using different PLMs, we conduct an ablation experiment with several PLMs, as shown in Table 3. The selection of PLMs influences the performance of our MER framework, primarily due to the fact that while all four PLMs are Transformer-based and pre-trained on large-scale corpora, their training strategies and corpora differ slightly. Notably, MER framework integrated with ERNIE, DeBERTa, or RoBERTa outperforms MER-BERT, as these three PLMs incorporate targeted optimizations over the original BERT architecture. Among these, MER-RoBERTa achieved the best performance. It removes the next sentence prediction objective, is pre-trained on an expanded dataset, and adopts adjusted key hyper-parameters. The details of PLMs and corresponding complete table can be found in Appendix F.

Parameter study of noise factor The noise factor ϵ introduces controlled perturbations to the hidden representations, serving as a regularizer to prevent router collapse. Figure 8 illustrates the model performance across various values of ϵ . We observe that $\epsilon = 0.1$ yields the best results. When $\epsilon = 0$, the lack of noise makes the model more prone to overfitting and leads to suboptimal utilization of experts. Conversely, when ϵ is too large, excessive noise distorts the original features and interferes with the selection of the router, which results in a drop in performance. The details of further experiments can be found in Appendix D.

3.4 Data Efficiency Study

Figure 9 illustrates the F1 scores of the MER model across four event relation types under varying few-shot scenarios. The results demonstrate a consistent performance improvement as the proportion of available training data increases from 0% to 100%. Notably, the model achieves a substantial performance leap when transitioning from the zero-shot

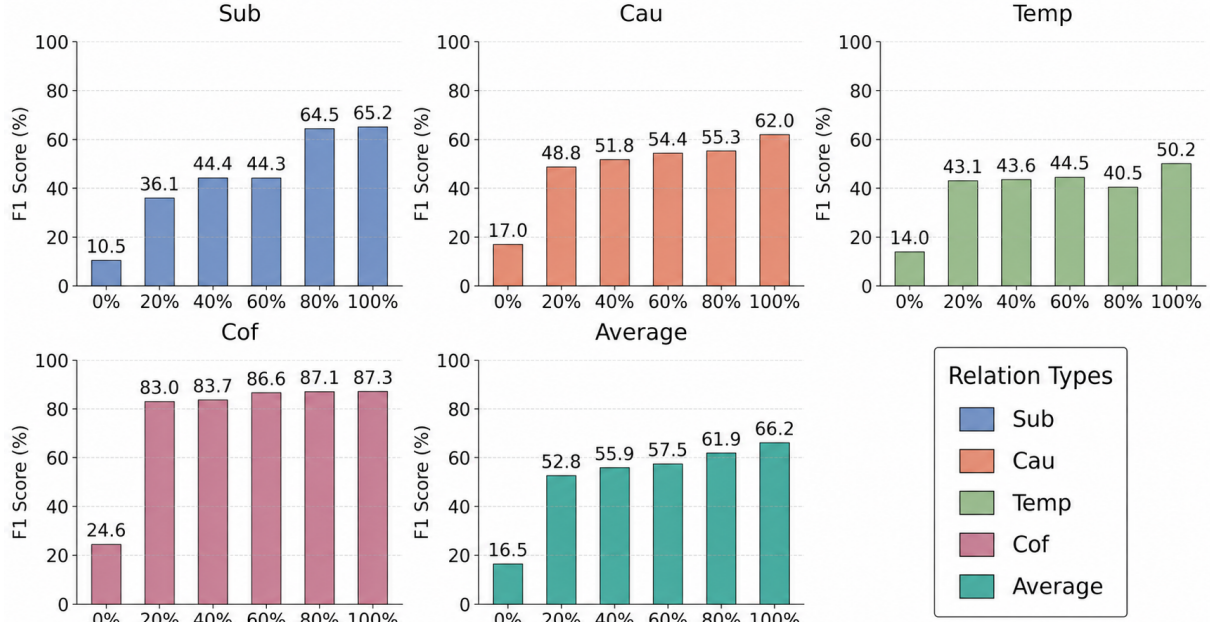


Figure 9: F1 scores of MER across different event relation types in different training-data proportions.

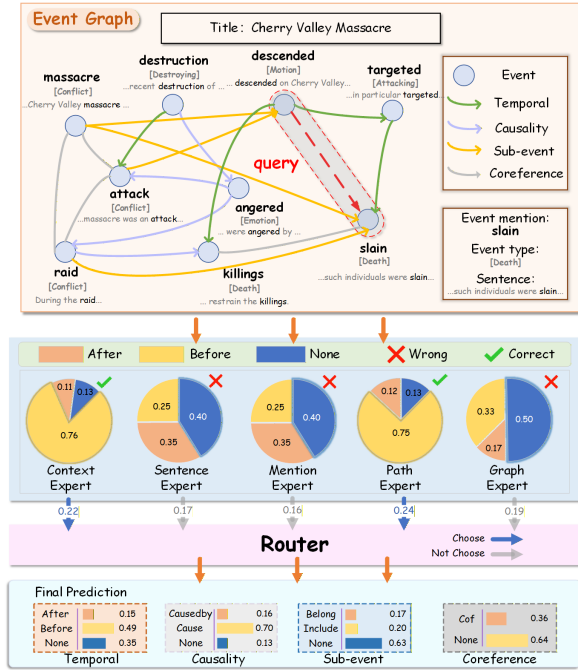


Figure 10: A case sample of our MER framework.

setting to the 20% setting. These results confirm the effectiveness of MER even with limited training data, benefiting from the abundant knowledge embedded in advanced PLMs. Furthermore, while PLMs typically struggle to capture task-specific structural correlations in the zero-shot setting, the rapid adaptation of MER with minimal data validates that our proposed architecture successfully activates this pre-trained knowledge and aligns it with the complex semantics of event graphs.

4 Case Study

Figure 10 presents a case study to illustrate the advantages of our MER model. In this instance, when querying the relation between the events "descended" and "slain", the Mention expert, Sentence expert, and Graph expert incorrectly predict a *None* relation, while the Context expert and Path expert successfully capture the ground-truth (*Before*) by leveraging deep contextual information from the document sentences and ordered relational paths between event pairs, respectively. The AER module adaptively identifies these two experts as the most relevant to the current query, and assigns them higher activation weights to aggregate their outputs into the final correct prediction of *Before*.

5 Conclusion

This paper has proposed the MER framework for the EGC task, which leverages a MoE architecture to capture both the semantic and structural information of event graphs. We design multiple specialized experts to handle distinct relational features by employing prompt learning and a graph attention network. An adaptive expert router is then introduced to dynamically select and fuse the predictions of the most relevant experts for each specific query event pair. Experiments on the EGC-MAVEN dataset demonstrate that MER outperforms existing methods. In future work, we plan to explore various load balancing strategies.

Limitations

Exploration of more expert design The specialized experts in our MER framework are manually designed based on human intuition. While effective, this hand-crafted approach may not capture all latent features of complex event graphs. In future work, we plan to explore a broader variety of experts to capture different levels of structural and semantic information. Additionally, we aim to investigate automated expert design methods (e.g., automated prompt search or architecture discovery) to reduce the reliance on manual engineering and further enhance the framework’s adaptability.

End-to-End Training Our framework currently adopts a two-stage training strategy, in which the experts and the router are optimized separately. As a result, the interactions between expert specialization and routing adaptation are not jointly optimized within a unified training process. Developing a stable end-to-end optimization framework that can effectively coordinate heterogeneous experts and the router remains an important direction for future work.

References

- Po-Kai Chen, Bo-Wei Tsai, Shao Kuan Wei, Chien-Yao Wang, Jia-Ching Wang, and Yi-Ting Huang. 2025. [Mixture of ordered scoring experts for cross-prompt essay trait scoring](#). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 18071–18084, Vienna, Austria. Association for Computational Linguistics.
- DeepSeek-AI, Aixin Liu, Aoxue Mei, Bangcai Lin, Bing Xue, Bingxuan Wang, Bingzheng Xu, Bochao Wu, Bowei Zhang, Chaofan Lin, Chen Dong, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenhao Xu, Chong Ruan, Damai Dai, Daya Guo, Dejian Yang, and 245 others. 2025. [DeepSeek-V3.2: Pushing the frontier of open large language models](#). Preprint, arXiv:2512.02556.
- Na Deng. 2021. Predicting social events using entity interaction graph sequences. In *2021 IEEE 2nd International Conference on Big Data, Artificial Intelligence and Internet of Things Engineering (ICBAIE)*, pages 1025–1029. IEEE.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. [BERT: Pre-training of deep bidirectional transformers for language understanding](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.
- Chenyang Du, Xiaoge Li, and Zhongyang Li. 2024. Semantic-enhanced reasoning question answering over temporal knowledge graphs. *Journal of Intelligent Information Systems*, 62(3):859–881.
- Li Du, Xiao Ding, Yue Zhang, Ting Liu, and Bing Qin. 2022. [A graph enhanced BERT model for event prediction](#). In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 2628–2638, Dublin, Ireland. Association for Computational Linguistics.
- Wensheng Gan, Zhenyao Ning, Zhenlian Qi, and Philip S. Yu. 2026. [Mixture of experts \(MoE\): A big data perspective](#). *Information Fusion*, 127:103664.
- Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 542 others. 2024. [The Llama 3 Herd of Models](#). Preprint, arXiv:2407.21783.
- Saiping Guan, Xueqi Cheng, Long Bai, Fujun Zhang, Zixuan Li, Yutao Zeng, Xiaolong Jin, and Jiafeng Guo. 2023. [What is event knowledge graph: A survey](#). *IEEE Trans. Knowl. Data Eng.*, 35(7):7569–7589.
- Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. [DeBERTa: Decoding-enhanced BERT with disentangled attention](#). In *International Conference on Learning Representations*.
- Shaonian Huang, Huanran Wang, Peilin Li, and Zhixin Chen. 2025. Document-level future event prediction integrating event knowledge graph and LLM temporal reasoning. *Electronics*, 14(19):3827.
- EunJeong Hwang, Jay-Yoon Lee, Tianyi Yang, Dhruvesh Patel, Dongxu Zhang, and Andrew McCallum. 2022. Event-event relation extraction using probabilistic box embedding. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 235–244.
- Xiaomeng Jin, Manling Li, and Heng Ji. 2022. Event schema induction with double graph autoencoders. In *Proceedings of the 2022 conference of the North American chapter of the Association for Computational Linguistics: human language technologies*, pages 2013–2025.
- Manling Li, Tengfei Ma, Mo Yu, Lingfei Wu, Tian Gao, Heng Ji, and Kathleen McKeown. 2021. Timeline summarization based on event graph compression via time-aware optimal transport. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6443–6456.

- Quan Li, Wenchao Yu, Suhang Wang, Minhua Lin, Lingwei Chen, Wei Cheng, and Haifeng Chen. 2025a. [xTime: Extreme event prediction with hierarchical knowledge distillation and expert fusion](#). *Preprint*, arXiv:2510.20651.
- Ruochen Li, Zimu Wang, and Xinya Du. 2025b. Efficient document-level event relation extraction. In *Proceedings of the 10th Workshop on Representation Learning for NLP (RepL4NLP-2025)*, pages 92–99.
- Chao Liang, Bang Wang, Chuanhong Zhan, and Wei Xiang. 2026. [Multiplex graph prompt learning and attentive fusion for event graph completion](#). *Neural Networks*, 199:108730.
- Qika Lin, Jun Liu, Fangzhi Xu, Yudai Pan, Yifan Zhu, Lingling Zhang, and Tianzhe Zhao. 2022. Incorporating context graph with logical reasoning for inductive relation prediction. In *Proceedings of the 45th international ACM SIGIR conference on research and development in information retrieval*, pages 893–903.
- Lihui Liu, Blaine Hill, Boxin Du, Fei Wang, and Hanghang Tong. 2024. Conversational question answering with language models generated reformulations over knowledge graph. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 839–850.
- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. [RoBERTa: A robustly optimized BERT pretraining approach](#). *arXiv preprint arXiv:1907.11692*.
- Yuting Liu, Ming Liu, Baowei Wang, Kun Ding, Shanshan Liu, and Liu Liu. 2023. [Script event prediction: Methods, evaluation, and challenges](#). *Application Research of Computers*, 40(5):1303–1311.
- Ka Man Lo, Zeyu Huang, Zihan Qiu, Zili Wang, and Jie Fu. 2025. [A closer look into mixture-of-experts in large language models](#). In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 4427–4447, Albuquerque, New Mexico. Association for Computational Linguistics.
- Rong Ma, Bo Ma, Zhen Wang, Aizimaiti Ainiwaer, Yating Yang, and Lei Wang. 2025. [Context-aware event forecast method based on semantic enhancement and candidate ranking optimization](#). *Application Research of Computers*, 42(9):2599–2606.
- Yunshan Ma, Chenchen Ye, Zijian Wu, Xiang Wang, Yixin Cao, and Tat-Seng Chua. 2023. Context-aware event forecasting via graph disentanglement. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 1643–1652.
- Saeed Masoudnia and Reza Ebrahimpour. 2014. Mixture of experts: a literature survey. *Artificial Intelligence Review*, 42(2):275–293.
- Bonan Min, Hayley Ross, Elior Sulem, Amir Pouran Ben Veyseh, Thien Huu Nguyen, Oscar Sainz, Eneko Agirre, Ilana Heintz, and Dan Roth. 2023. Recent advances in natural language processing via large pre-trained language models: A survey. *ACM Computing Surveys*, 56(2):1–40.
- Siyuan Mu and Sen Lin. 2025. [A comprehensive survey of mixture-of-experts: Algorithms, theory, and applications](#). *Preprint*, arXiv:2503.07137.
- OpenAI. 2024. [GPT-4o mini: Advancing cost-efficient intelligence](#). Official model announcement.
- Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F. Christiano, Jan Leike, and Ryan Lowe. 2022. [Training language models to follow instructions with human feedback](#). In *Advances in Neural Information Processing Systems*, volume 35, pages 27730–27744.
- Zhen Qin, Yicheng Cheng, Zhe Zhao, Zhe Chen, Donald Metzler, and Jingzheng Qin. 2020. Multitask mixture of sequential experts for user activity streams. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 3083–3091.
- Paolo Rosso, Dingqi Yang, and Philippe Cudré-Mauroux. 2020. Beyond triplets: hyper-relational knowledge graph embedding for link prediction. In *Proceedings of the web conference 2020*, pages 1885–1896.
- Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc V. Le, Geoffrey Hinton, and Jeff Dean. 2017. [Outrageously Large Neural Networks: The Sparsely-Gated Mixture-of-Experts Layer](#). In *International Conference on Learning Representations*.
- Lei Shi, Bin Hu, Deng Zhao, Jianshan He, Zhiqiang Zhang, and Jun Zhou. 2024. Structural information enhanced graph representation for link prediction. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pages 14964–14972.
- Zhixiang Su, Di Wang, Chunyan Miao, and Lizhen Cui. 2024. Anchoring path for inductive relation prediction in knowledge graphs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 9011–9018.
- Jialong Tang, Hongyu Lin, Meng Liao, Yaojie Lu, Xi-anpei Han, Le Sun, Weijian Xie, and Jin Xu. 2021. From discourse to narrative: Knowledge projection for event relation extraction. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 732–742.
- Team GLM. 2024. [ChatGLM: A family of large language models from GLM-130B to GLM-4 all tools](#). *Preprint*, arXiv:2406.12793.

- Takanori Ugai, Shusaku Egami, and Ken Fukuda. 2024. Event prediction in event-centric knowledge graphs using BERT. In *2024 IEEE 18th International Conference on Semantic Computing (ICSC)*, pages 306–310. IEEE.
- Hongwei Wang, Zixuan Zhang, Sha Li, Jiawei Han, Yizhou Sun, Hanghang Tong, Joseph P. Olive, and Heng Ji. 2022a. [Schema-guided event graph completion](#). In *4th Conference on Automated Knowledge Base Construction*.
- Lihong Wang, Juwei Yue, Shu Guo, Jiawei Sheng, Qianren Mao, Zhenyu Chen, Shenghai Zhong, and Chen Li. 2021. Multi-level connection enhanced representation learning for script event prediction. In *Proceedings of the web conference 2021*, pages 3524–3533.
- Xiaozhi Wang, Yulin Chen, Ning Ding, Hao Peng, Zimu Wang, Yankai Lin, Xu Han, Lei Hou, Juanzi Li, Zhiyuan Liu, Peng Li, and Jie Zhou. 2022b. [MAVEN-ERE: A unified large-scale dataset for event coreference, temporal, causal, and subevent relation extraction](#). In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 926–941, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.
- Wei Xiang, Chao Liang, and Bang Wang. 2023. [TEPrompt: Task enlightenment prompt learning for implicit discourse relation recognition](#). In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 12403–12414, Toronto, Canada. Association for Computational Linguistics.
- Danyang Zhang, Junhao Song, Ziqian Bi, Xinyuan Song, Yingfang Yuan, Tianyang Wang, Joe Yeong, and Junfeng Hao. 2025. [Mixture of experts in large language models](#). Preprint, arXiv:2507.11181.
- Zhengyan Zhang, Xu Han, Zhiyuan Liu, Xin Jiang, Maosong Sun, and Qun Liu. 2019. [ERNIE: Enhanced language representation with informative entities](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1441–1451, Florence, Italy. Association for Computational Linguistics.
- Pengpeng Zhou, Bin Wu, Caiyong Wang, and Longzhu He. 2025. An improved hierarchical neural network model with local and global feature matching for script event prediction. *Expert Systems with Applications*, 259:125325.
- Yanqi Zhou, Tao Lei, Hanxiao Liu, Nan Du, Yanping Huang, Vincent Zhao, Andrew M. Dai, Zhifeng Chen, Quoc V. Le, and James Laudon. 2022. [Mixture-of-experts with expert choice routing](#). In *Advances in Neural Information Processing Systems*, volume 35, pages 7103–7114.
- Xun Zhu, Wang Gao, Tianyu Li, Wenguang Yao, and Hongtao Deng. 2024. Event-centric hierarchical hyperbolic graph for multi-hop question answering over knowledge graphs. *Engineering Applications of Artificial Intelligence*, 133:107971.

A Related Work

A.1 Event Prediction and Completion

The task of event prediction is to predict future events based on the content and relationships of historical events, which provides decision-making support for potential future developments (Deng, 2021; Wang et al., 2021; Du et al., 2022; Liu et al., 2023; Ma et al., 2023, 2025). For instance, event knowledge graphs and large language model reasoning frameworks are integrated to enhance temporal reasoning and extract event dependencies for predicting document-level future events (Huang et al., 2025). An improved hierarchical neural network is proposed, which integrates local and global feature matching to capture event context information for event prediction (Zhou et al., 2025).

Besides, the Event Graph Completion task is introduced, which aims to predict missing event nodes for event graphs. They use event schema to describe the stereotypical structure of event graphs, while incorporating neighbor and path information to capture high-order topological and semantic dependencies in the event schema graph for event node completion (Wang et al., 2022a).

A.2 Mixture of Experts

The Mixture of Experts architecture utilizes a mixture of specialized sub-models named experts to solve a task, which has been widely applied in various domains (Shazeer et al., 2017; Zhang et al., 2025; Lo et al., 2025; Gan et al., 2026; Mu and Lin, 2025). A mixture of experts model is leveraged to handle diverse and sequential user activity data by assigning different experts to different types of user activities (Qin et al., 2020). A mixture of experts mechanism is introduced, which dynamically selects and fuses outputs from expert models across different rarity levels to predict extreme events (Li et al., 2025a).

We argue that event graphs contain multi-level information, including entities, context, relationships, title, and other relevant details. To this end, our approach explores the use of the MoE framework to capture both semantic and structural information of EGs.

B Experiment Settings

Dataset: The statistics of EGC-MAVEN dataset are summarized in Table 7. This large-scale dataset comprises 3,614 complex event graphs and 18,181 event completion instances, with an average of

over 25 nodes and 60 edges per event graph. Further statistics on the distribution of event graph nodes and edges are presented in Table 4 and 5. Note that the majority of event graphs in EGC-MAVEN dataset exhibit structural complexity, with most graphs containing more than 10 nodes and over 20 edges. Besides, 5,543 instances in EGC-MAVEN dataset contain more than 30 nodes, and 5,203 instances involve event graphs with more than 70 edges. The distributions demonstrate that EGC-MAVEN dataset contains complex, large-scale event graphs, which aligns well with the requirements of real-world application scenarios.

Details of experiment setup: The parameter settings of proposed MER framework are shown in Table 6.

Metrics: We report the evaluation metrics in detail here for completeness. Specifically, we first calculate the macro Precision, Recall, and F1-score for each type of relations, which are formulated as binary or three-class classification tasks, to evaluate the performance of our model on a specific relation type. Then we calculate the average results of Precision, Recall, and F1-score to evaluate the overall performance of our model among all relation types, defined as follows:

$$P = \frac{1}{N} \sum_{\tau \in \mathcal{F}} P^\tau$$

$$R = \frac{1}{N} \sum_{\tau \in \mathcal{F}} R^\tau$$

$$F_1 = \frac{1}{N} \sum_{\tau \in \mathcal{F}} F_1^\tau$$

where P^τ , R^τ , and F_1^τ are the evaluation metrics for relations of each type τ and N is the number of types.

C Baselines

We compare the proposed MER framework with the following baselines:

HINGE learns directly from hyper-relational facts in knowledge graphs to capture both the primary structural information within triplets and the correlations between each triplet and its associated key-value pairs.

ConGLR jointly leverages a context graph and logical rules, which integrates graph structural patterns with symbolic reasoning to predict new relations.

Split	(0,5]	(5,10]	(10,15]	(15,20]	(20,25]	(25,30]	(30,+∞)
Train	120	1,000	1,884	2,018	1,462	1,201	3,306
Val.	36	297	547	647	515	422	1,125
Test	21	380	571	704	515	298	1,112
Total	177	1,677	3,002	3,369	2,492	1,921	5,543

Table 4: Distribution of event graph nodes.

Split	(0,10]	(10,20]	(20,30]	(30,40]	(40,50]	(50,60]	(60,70]	(70,+∞)
Train	1,053	1,658	1,563	1,251	1,012	803	574	3,077
Val.	274	459	472	416	365	320	250	1,033
Test	282	523	484	373	324	278	244	1,093
Total	1,609	2,640	2,519	2,040	1,701	1,401	1,068	5,203

Table 5: Distribution of event graph edges.

Parameter	Value
Top- k	2
Temperature (t)	0.9
Max Context Length	512
Max Sentence Schema Length	200
Max Mention Schema Length	75
Max Context Schema Length	285
Optimizer	AdamW
Expert Learning Rate	5e-6
Router Learning Rate	5e-6
Weight Decay	0.01
Epochs	15
Loss Function	Cross Entropy
Random Seed	209
noise factor ϵ	0.1

Table 6: Parameter settings for MER framework.

SIEG is a structure-oriented link prediction framework that reduces reliance on neighbor node features and employs a Binary Structural Transformer to encode structural relationships between target nodes.

APST supports inductive relation prediction by jointly modeling Anchoring Paths and Closed Paths with a Sentence Transformer. It retrieves entity descriptions to construct informative paths when closed paths are unavailable.

PLAF integrates Dual Graph Prompt Learning with a Multiplex Graph Attention Network by reformulating event graphs into triplet sequences to capture both semantic and structural information.

ChatGLM3-6B is a bilingual conversational model developed by Zhipu AI¹ and Tsinghua University KEG Lab², built upon the General Language Model (GLM) architecture. It utilizes large-

scale pre-training on diverse multilingual datasets and fine-tuning to optimize performance in generating contextually accurate responses.

Llama3-8B is an open-source large language model released by Meta³, which is based on the Transformer decoder architecture and utilizes supervised fine-tuning and reinforcement learning from human feedback during its training process.

GPT-3.5-turbo is a conversational AI model developed by OpenAI⁴, which utilizes a generative pre-training framework, followed by instruction tuning and reinforcement learning from human feedback to ensure alignment with user intent.

GPT-4o-mini is a cost-effective multimodal model developed by OpenAI⁵. Built upon the GPT-4o architecture, it is optimized for low-latency inference and utilizes instruction tuning to deliver robust reasoning capabilities with significantly reduced computational overhead.

DeepSeek-V3.2 is a highly efficient large language model developed by DeepSeek⁶, utilizing a Mixture-of-Experts (MoE) architecture to deliver strong reasoning and language understanding capabilities with minimal computational overhead.

For LLMs, we use prompt learning methods to convert event graphs into text sequences, constructing prompt templates to input into the LLM. Specifically, we input the description of the event graph, context sentences, linearized event graph structure, and queries as prompts into the LLM to complete the relationships in the event graph. The prompt is shown in Figure 3.

D Further experiments

D.1 Ablation study of the Path Expert

To investigate how to effectively integrate event mentions and types within the Path Expert, we compare four template strategies: (1) Only Mention: uses only the mention path template; (2) Only Type: uses only the type path template; (3) Combined: forms a single prompt by appending the type sequence directly after the mention sequence; (4) Fused (i.e., our default Path Expert): uses the two templates independently.

As shown in Table 8, the Fused strategy achieves the best average F1-score. Notably, the Combined strategy experiences a significant perfor-

¹<https://www.modelscope.cn/models/ZhipuAI/chatglm3-6b>

²<https://github.com/THUDM/ChatGLM-6B>

³<https://github.com/meta-llama/llama3>

⁴<https://openai.com/>

⁵<https://openai.com/>

⁶<https://huggingface.co/deepseek-ai/DeepSeek-V3.2>

Split	Graphs	Instances	Avg. Nodes	Avg. Edges	sub-event		Temporal		Causality		Corf.
					Belong	Include	Before	After	Cause	Caused_by	
Train	2,198	10,991	26	60	567	557	1,474	1,478	1,322	1,220	1,289
Val.	708	3,589	27	66	181	182	477	453	427	407	457
Test	708	3,601	26	64	213	193	495	474	425	405	426
Total	3,614	18,181	–	–	961	932	2,446	2,405	2,174	2,032	2,172

Table 7: Statistics of the EGC-MAVEN dataset.

Fusion Strategy	Sub-event			Causality			Temporal			Coreference			Average		
	R	P	F1	R	P	F1	R	P	F1	R	P	F1	R	P	F1
Only Mention	<u>66.1</u>	<u>39.1</u>	<u>49.1</u>	49.4	<u>34.2</u>	40.4	45.1	<u>26.3</u>	<u>32.9</u>	92.5	77.1	<u>84.1</u>	63.3	<u>44.2</u>	<u>51.6</u>
Only Type	49.5	23.8	32.0	65.1	24.3	35.4	50.7	22.4	31.1	99.1	66.9	79.9	<u>66.1</u>	34.4	44.6
Combined	62.8	37.7	47.0	55.5	32.6	<u>41.0</u>	44.1	25.7	32.5	97.4	<u>77.0</u>	86.0	64.9	43.2	<u>51.6</u>
Fused (Ours)	73.0	56.3	63.5	<u>64.6</u>	46.4	54.0	<u>48.0</u>	34.3	40.0	<u>98.8</u>	69.1	81.4	71.1	51.5	59.7

Table 8: Ablation study of the Path Expert on template fusion strategies. The best results are in bold, and the sub-optimal results are underlined.

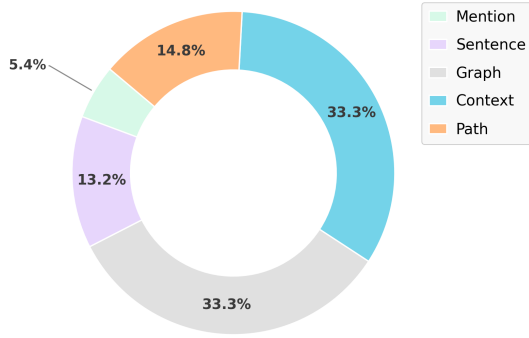


Figure 11: Distribution of experts on train set with top- k set to 3.

mance drop, yielding no improvement over the Only Mention strategy. This indicates that naively inserting mention types into the prompt is ineffective, as PLMs struggle to allocate sufficient attention to this appended information. Consequently, our decoupled strategy processes the mention and type paths independently, preserving their distinct semantic properties without mutual interference. This comparison validates the necessity of our dual-template fusion mechanism for the Path Expert.

D.2 Distribution of different experts

Figure 11 and Figure 12 illustrate the distribution of expert selection on the training and validation sets with top- k set to 3, respectively. Consistent with the distribution of different experts on the test set, the router favors the Context and Path experts, which indicates that contextual and path-level information are crucial for EGC. It is interesting to

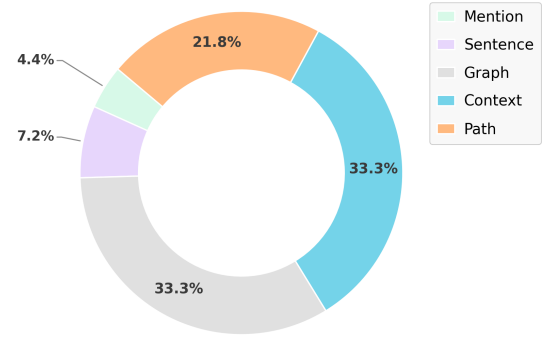


Figure 12: Distribution of experts on valid set with top- k set to 3.

note that the Mention expert, Sentence expert, and Graph expert are not entirely discarded but are adaptively activated to handle some event graphs.

E Prompts for different experts

Figure 13 illustrates the prompt templates of different experts, which are designed to transform event graphs into sequential textual inputs for the PLMs. Specifically, Q denotes the target query event pair to be predicted, and $\langle \text{Rel} \rangle$ represents the relationship connecting the intermediate event nodes along the path. By converting the graph structure into a text sequence ending with a [Mask] token, we employ the MLM head to directly predict the event relation.

F Pre-trained Language Models

Pre-trained language models are introduced as follows:

Event Mention Template:

According to the common knowledge, [Mention]-Q-1 and [Mention]-Q-2 is [Mask].

Event Sentence Template:

[Title], [Sentence-Q-1, Sentence-Q-2], [Mention]-Q-1 and [Mention]-Q-2 is [Mask].

Event Context Template:

[Title], [Sentence-Q-1, ...(Sentences between Query event pair), Sentence-Q-2], [Mention]-Q-1 and [Mention]-Q-2 is [Mask].

Event Mention Path Template:

[Mention]-P-1 <Rel> [Mention]-P-2 </s> [Mention]-P-2 <Rel> [Mention]-P-3 </s> (Mention Path Triplet) </s> [Mention]-Q-1 [Mask] [Mention]-Q-2.

Event Type Path Template:

[Type]-P-1 <Rel> [Type]-P-2 </s> [Type]-P-2 <Rel> [Type]-P-3 </s> (Type Path Triplet) </s> [Type]-Q-1 [Mask] [Type]-Q-2.

Figure 13: Illustration of the prompts for different experts.

BERT (Devlin et al., 2019) is a pioneering model that utilizes a masked language modeling (MLM) objective to pre-train deep bidirectional representations from unlabeled text.

RoBERTa (Liu et al., 2019) optimizes the BERT architecture by removing the next-sentence prediction (NSP) objective and training with larger batches and more data on longer sequences with dynamic masking.

DeBERTa (He et al., 2021) enhances BERT by employing a disentangled attention mechanism that separately encodes content and positional information, along with an enhanced mask decoder to incorporate absolute positions.

ERNIE (Zhang et al., 2019) is designed to integrate external knowledge into the pre-training process by masking phrases and entities, thereby capturing semantic relationships more effectively.

The full results are presented in Table 9.

G Analysis of computational cost

Table 10 reports the inference time, GPU memory usage, and computational complexity of different modules. Here, S denotes the input sequence length, d the hidden dimension, V the vocabulary size, N and E the numbers of nodes and edges in the input event graph, respectively, L the number of Transformer layers, and K the number of activated

experts.

The results indicate that MER is computationally efficient and can be trained with moderate hardware resources. MER adopts a two-stage training strategy that separates expert training from router training, which helps reduce the memory overhead and optimization burden associated with jointly training multiple experts. During inference, the router selects only the top- K most relevant experts for weighted fusion, thereby limiting the additional computational cost introduced by the multi-expert architecture. In practice, both training and inference can be performed on a single NVIDIA RTX 3090 GPU within a reasonable amount of time. These results demonstrate that MER achieves good computational efficiency and practical feasibility.

H Further Analysis of Expert Ablations

The ablation results in Table 2 show that removing individual experts consistently degrades the overall performance of MER, suggesting that these experts provide complementary information for heterogeneous relation prediction. The Path Expert exhibits a particularly pronounced contribution to structurally dependent relations. Removing it decreases the F1 scores on Sub-event and Causality from 65.2 to 49.8 and from 62.0 to 55.7, corresponding to drops of 15.4 and 6.3 percentage

Model	Module	Sub-event			Causality			Temporal			Coreference			Average		
		R	P	F1	R	P	F1	R	P	F1	R	P	F1	R	P	F1
MER BERT	Mention	45.7	13.7	20.8	49.1	19.3	27.6	42.4	19.9	27.0	82.2	34.8	48.9	54.9	21.9	31.1
	Sentence	55.0	30.9	39.4	51.6	37.2	42.9	33.2	32.3	32.2	83.3	75.1	79.0	55.8	43.9	48.4
	Context	60.2	36.2	45.2	64.6	40.3	49.6	53.6	35.2	42.5	92.0	74.8	82.5	67.6	46.6	55.0
	Path	70.4	64.2	67.2	59.8	44.8	51.0	43.5	30.9	35.8	97.0	75.5	84.9	67.7	53.9	59.7
	Graph	76.6	22.6	34.8	68.8	28.4	40.2	47.6	23.7	31.6	76.3	69.6	72.8	67.3	36.1	44.9
	Router	70.4	55.0	61.8	59.9	45.1	51.2	43.4	31.0	36.2	97.0	75.8	85.1	67.7	51.7	58.6
MER ERNIE	Mention	48.9	13.9	21.5	46.2	19.4	27.3	32.2	19.8	24.0	79.8	35.1	48.7	51.8	22.1	30.4
	Sentence	45.4	32.3	37.7	55.0	36.2	43.5	42.5	34.6	38.1	80.1	60.9	69.2	55.8	41.0	47.1
	Context	64.6	36.2	46.3	65.0	43.6	52.1	51.9	40.7	45.6	92.5	78.2	84.7	68.5	49.7	57.2
	Path	73.9	57.7	64.8	63.3	46.6	53.7	47.5	32.0	37.6	98.8	68.5	80.9	70.9	51.2	59.3
	Graph	78.0	18.7	30.2	78.0	25.4	38.4	50.8	24.2	32.8	60.6	70.1	65.0	66.9	34.6	41.6
	Router	66.5	61.2	63.8	64.2	51.0	56.9	46.4	38.7	42.1	96.7	81.1	88.2	68.5	58.0	62.8
MER DeBERTa	Mention	43.4	14.1	21.2	47.4	19.3	27.4	41.6	19.3	26.2	82.4	34.3	48.4	53.7	21.8	30.8
	Sentence	57.2	40.0	47.1	63.3	46.1	53.3	48.9	34.9	41.0	93.9	77.7	86.0	65.8	49.7	56.9
	Context	53.8	42.6	47.6	69.5	47.0	56.1	57.2	39.2	46.5	93.4	82.2	87.5	68.5	52.8	59.4
	Path	74.3	61.4	66.4	56.6	43.2	49.0	52.7	29.0	37.4	99.3	66.8	79.9	70.7	50.1	58.2
	Graph	72.7	21.8	33.5	66.6	27.1	38.5	48.6	23.7	31.8	79.6	68.1	73.4	66.9	35.2	44.3
	Router	59.2	45.7	51.6	71.7	47.7	57.2	57.2	38.0	45.6	94.8	79.8	86.7	70.7	52.8	60.3
MER RoBERTa	Mention	41.5	14.9	21.3	53.2	20.6	29.5	45.0	19.0	26.6	84.0	33.6	48.0	55.9	22.0	31.4
	Sentence	66.3	35.7	46.4	73.2	37.3	49.4	48.5	34.0	39.8	97.7	69.7	81.3	71.4	44.2	54.2
	Context	65.1	40.3	49.5	79.4	42.6	55.4	62.1	41.0	49.3	93.7	79.3	85.9	75.1	50.8	60.0
	Path	73.0	56.3	63.5	64.6	46.4	54.0	48.0	34.3	40.0	98.8	69.1	81.4	71.1	51.5	59.7
	Graph	80.4	20.0	32.0	71.2	26.8	38.9	52.5	22.7	31.7	78.6	64.9	71.1	70.7	33.6	43.4
	Router	73.0	59.1	65.2	76.6	52.1	62.0	59.0	42.8	50.2	97.0	79.4	87.3	76.4	58.4	66.2

Table 9: PLM Ablation results on the EGC-MAVEN dataset.

Module	Time Complexity	GPU Memory Usage	Training/Inference Time
Mention Expert	$\mathcal{O}(L(S^2d + Sd^2) + dV)$	~ 6 GB	25 / 2 min
Sentence Expert	$\mathcal{O}(L(S^2d + Sd^2) + dV)$	~ 6 GB	25 / 2 min
Context Expert	$\mathcal{O}(L(S^2d + Sd^2) + dV)$	~ 8 GB	25 / 2 min
Path Expert	$\mathcal{O}(2L(S^2d + Sd^2) + d^2)$	~ 10 GB	45 / 3 min
Graph Expert	$\mathcal{O}(NL(S^2d + Sd^2) + Nd^2 + Ed)$	~ 12 GB	46 / 3 min
Router	$\mathcal{O}(L(S^2d + Sd^2) + dK)$	~ 6 GB	40 / 6 min

Table 10: Computational cost analysis of different modules in MER.

points, respectively. Performance also decreases on Temporal and Coreference relations by 1.0 and 1.5 points, leading to an overall reduction of 6.1 points in average F1. These results indicate that the Path Expert is particularly important for capturing structural dependencies underlying Sub-event and Causality relations, while also providing complementary information for Temporal and Coreference prediction.

Table 11 presents representative cases to further analyze the effects observed in the expert ablation study. In the Temporal case, the path from **planned**

to **attack** captures the progression from planning to execution, providing evidence for their temporal order. In the Causality case, intermediate events connect **attack** with **set up**, offering multi-hop evidence for the causal relation. In contrast, the Coreference case is determined more directly by the semantic and contextual consistency between **conflict₁** and **conflict₂**, while the surrounding path provides limited additional evidence. These cases help explain why the Path Expert contributes more strongly to Temporal and Causality relations than to Coreference, while still improving the overall

Relation	Query Pair	Label	Representative Event Path	Role of Path Evidence
Temporal	attack vs. planned	After	proposed → planned → decided → attempt → attack	The intermediate events describe the progression from planning to execution, providing structural evidence that planned occurs before attack .
Causality	set up vs. attack	Caused_by	attack → started → fell → set up → ruled → announced	The path connects the earlier attack with the subsequent set up event through intermediate events, providing useful multi-hop evidence for causal reasoning.
Coreference	conflict ₁ vs. conflict ₂	Coreference	conflict ₁ → capture ← conflict ₂ → operations → gained	The surrounding events characterize the broader event context but provide limited additional evidence for determining whether the two mentions refer to the same event.

Table 11: Representative cases for further analysis of the expert ablation results.

Model	Shot	Sub.			Cau.			Tem.			Corf.			Avg.		
		R	P	F1	R	P	F1	R	P	F1	R	P	F1	R	P	F1
GPT-3.5	0	38.7	12.0	18.2	47.6	14.6	22.3	41.0	18.4	24.7	89.4	20.5	33.4	54.2	16.4	24.6
GPT-3.5	1	45.8	27.5	25.4	41.0	24.4	29.7	43.4	48.1	35.8	41.4	30.4	31.4	42.9	32.6	30.6
GPT-3.5	3	42.8	47.1	34.4	42.6	33.3	34.9	42.8	47.1	34.4	41.2	29.0	32.8	42.4	39.1	34.1
GPT-3.5	5	46.3	34.5	37.3	43.4	37.2	35.0	41.0	42.3	37.8	42.6	33.6	36.6	43.3	36.9	36.7
GPT-4o-mini	0	46.0	25.2	22.7	35.9	16.9	22.9	41.2	40.9	39.0	42.7	38.5	38.0	41.5	30.4	30.7
GPT-4o-mini	1	43.3	21.8	19.7	36.1	17.1	21.9	42.3	43.2	33.4	22.1	13.2	10.4	36.0	23.8	21.4
GPT-4o-mini	3	46.8	33.3	25.6	36.4	16.4	22.2	43.9	44.8	35.6	27.8	16.0	14.6	38.7	27.6	24.5
GPT-4o-mini	5	45.8	29.4	27.1	38.7	22.9	25.3	41.7	29.4	34.5	26.3	12.8	12.4	38.1	23.6	24.8
DeepSeek-V3.2	0	48.2	19.0	22.2	34.9	12.4	17.3	42.8	31.2	33.8	40.0	34.7	30.3	41.5	24.3	25.9
DeepSeek-V3.2	1	52.6	33.2	32.8	33.6	14.1	18.7	38.1	28.5	31.5	27.9	17.3	15.3	38.1	23.3	24.6
DeepSeek-V3.2	3	60.0	40.6	44.1	32.4	12.0	16.4	40.9	32.5	32.1	40.3	31.5	31.8	43.4	29.2	31.1
DeepSeek-V3.2	5	67.7	34.1	45.1	36.0	17.8	22.2	39.8	35.1	36.0	34.1	18.6	20.3	44.4	26.4	30.9
MER (ours)	–	73.0	59.1	65.2	76.6	52.1	62.0	59.0	42.8	50.2	97.0	79.4	87.3	76.4	58.4	66.2

Table 12: Comparison with LLMs under different few-shot ICL settings.

performance of MER.

I Further Comparison with Few-shot ICL

To further examine whether in-context demonstrations can improve the performance of general-purpose LLMs on EGC, we evaluate GPT-3.5, GPT-4o-mini, and DeepSeek-V3.2 under 0-, 1-, 3-, and 5-shot settings. For each relation type t , we define its candidate label set $\mathcal{Y}_t$ and sample demonstrations from the training set. For 1-shot prompting, one demonstration is randomly selected. For 3- and 5-shot prompting, we first select examples to cover the candidate labels and then randomly sample additional examples when necessary until the required number is reached. The selected demonstrations are placed before the test query, and the model is instructed to predict one label from $\mathcal{Y}_t$.

As shown in Table 12, increasing the number

of demonstrations does not consistently improve performance across different LLMs. For example, GPT-4o-mini achieves an average F1 of 30.7 in the zero-shot setting, while its 1-, 3-, and 5-shot results are 21.4, 24.5, and 24.8, respectively. DeepSeek-V3.2 also exhibits fluctuations as the number of demonstrations increases. Although GPT-3.5 benefits more from additional examples and reaches its best average F1 of 36.7 with five demonstrations, it remains substantially below MER (66.2). These results indicate that few-shot demonstrations alone are insufficient to reliably capture the fine-grained semantic and structural distinctions among heterogeneous event relations. In contrast, MER explicitly models complementary relation information through specialized experts and achieves more consistent performance across relation types.